It's all about Java and Liferay

Home   Privacy Policy   About Myself

Saturday, May 26, 2012

Connecting to different database using Liferay Service Builder

Generally we guys used to connect with the liferay default Database.
We also have requirement that Client wants there Data on Separate Schema or Separate DB.
But In Liferay how we can connect with different DB?

Don't Worry guys your problem is now solved.

Just Follow some steps as below :-

1) As you are mentioning the jdbc properties for the liferayportal in portal-ext.properties, same way mention for the external DB or seperate schema as below :-

```
1   jdbc.default.url=jdbc\:mysql\://localhost/liferay612?useUnicode\=true&characterE
2   jdbc.default.driverClassName=com.mysql.jdbc.Driver
3   jdbc.default.username=root
4   jdbc.default.password=root
5
6   jdbc.test.driverClassName=com.mysql.jdbc.Driver
7   jdbc.test.url=jdbc:mysql://localhost/testE?useUnicode=true&characterEncoding=UTF
8   jdbc.test.username=root
9   jdbc.test.password=root
```

Different DB portal-ext.properties hosted with ❤ by **GitHub**   **view raw**

Explanation :-
The first four line starts with the **jdbc.default**, it means that it was associated with the Default Data Source. So it will connect to the Default Liferay DB.

Last four line line starts with the **jdbc.test**, which does not belong to the Default DB. It will be connected to some other DB whose entry will be in ext-spring.xml.

2) As we are using service-builder, it means that you need new tables other than liferay default DB. So it requires for you to create new plugins project and in that you need to create service.xml under webapps/WEB-INF and with the help of ANT(ant build-service) or MAVEN (mvn liferay:build-service) you will able to create full structure for your service. But still it is pointing to the default DB.

Now you need to create a new file ext-spring.xml under WEB-INF/src/META-INF dir. Inside META-INF folder you will find couple of xml files whose entry will be there in liferay portal.properties file as below :-

```
1    spring.configs=\
2          META-INF/base-spring.xml,\
3          \
4          META-INF/hibernate-spring.xml,\
5          META-INF/infrastructure-spring.xml,\
6          META-INF/management-spring.xml,\
7          \
8          META-INF/util-spring.xml,\
9          \
10         META-INF/jpa-spring.xml,\
11         \
```

```
12          META-INF/executor-spring.xml,\
13          \
14          META-INF/audit-spring.xml,\
15          META-INF/cluster-spring.xml,\
16          META-INF/editor-spring.xml,\
17          META-INF/jcr-spring.xml,\
18          META-INF/ldap-spring.xml,\
19          META-INF/messaging-core-spring.xml,\
20          META-INF/messaging-misc-spring.xml,\
21          META-INF/mobile-device-spring.xml,\
22          META-INF/notifications-spring.xml,\
23          META-INF/poller-spring.xml,\
24          META-INF/rules-spring.xml,\
25          META-INF/scheduler-spring.xml,\
26          META-INF/scripting-spring.xml,\
27          META-INF/search-spring.xml,\
28          META-INF/workflow-spring.xml,\
29          \
30          META-INF/counter-spring.xml,\
31          META-INF/mail-spring.xml,\
32          META-INF/portal-spring.xml,\
33          META-INF/portlet-container-spring.xml,\
34          META-INF/staging-spring.xml,\
35          META-INF/virtual-layouts-spring.xml,\
36          \
37          #META-INF/dynamic-data-source-spring.xml,\
38          #META-INF/shard-data-source-spring.xml,\
39          #META-INF/memcached-spring.xml,\
40          #META-INF/monitoring-spring.xml,\
41          \
42          META-INF/ext-spring.xml
```

Different DB spring properties hosted with ❤ by **GitHub**   **view raw**

If you notice the order of xml file loading in portal.properties file, then you will find that the last file is ext-spring.xml is loaded. So we will now **create ext-spring.xml** and putting all transaction,datasource and sessionfactory related changed on that file as below :-

```
1    <?xml version="1.0"?>
2    <beans xmlns="http://www.springframework.org/schema/beans"
3        xmlns:aop="http://www.springframework.org/schema/aop" xmlns:context="http:/
4            xmlns:tx="http://www.springframework.org/schema/tx" xmlns:xsi="http://w
5            xsi:schemaLocation="http://www.springframework.org/schema/aop http://ww
6
7            <aop:config>
8                    <aop:pointcut id="transactionOperation" expression="bean(*Servi
9                    <aop:advisor advice-ref="transactionAdvice" pointcut-ref="trans
10           </aop:config>
11
12           <bean id="basePersistence" abstract="true">
13                   <property name="dataSource" ref="anotherDataSource" />
14                   <property name="sessionFactory" ref="anotherSessionFactory" />
15           </bean>
16
17           <bean id="transactionAdvice" class="org.springframework.transaction.int
18                   <property name="transactionManager" ref="anotherTransactionMana
19                   <property name="transactionAttributeSource">
20                           <bean class="org.springframework.transaction.annotatior
21                               <constructor-arg>
22                                       <bean class="com.liferay.portal.spring.
23                               </constructor-arg>
24                           </bean>
25                   </property>
```

```
26      </bean>
27
28      <bean id="anotherHibernateSessionFactory" class="com.liferay.portal.spr
29          <property name="dataSource" ref="anotherDataSource" />
30      </bean>
31
32      <bean id="anotherSessionFactory" class="com.liferay.portal.dao.orm.hibe
33          <property name="sessionFactoryImplementor" ref="anotherHibernat
34      </bean>
35
36      <bean id="anotherTransactionManager" class="org.springframework.orm.hib
37          <property name="dataSource" ref="anotherDataSource" />
38          <property name="globalRollbackOnParticipationFailure" value="fa
39          <property name="sessionFactory" ref="anotherHibernateSessionFac
40      </bean>
41
42      <bean id="anotherDataSource" class="org.springframework.jdbc.datasource
43          <property name="targetDataSource">
44              <bean class="com.liferay.portal.dao.jdbc.util.DataSour
45                  <property name="propertyPrefix" value="jdbc.tes
46              </bean>
47          </property>
48      </bean>
49  </beans>
```

Different DB ext-spring.xml hosted with ❤ by **GitHub**   **view raw**

3) Now the last and the final step you need to do is that you need to modify the existing service.xml as follows :-

```
1  <service-builder package-path="com.test.liferay">
2  <namespace>test</namespace>
3  <entity data-source="anotherDataSource" local-service="true" name="testDB" remot
4  ......
5  ......
6  </entity>
7  </service-builder>
```

Different DB service.xml hosted with ❤ by **GitHub**   **view raw**

The **data-source** value specifies the data source target that is set to the persistence class. The default value is the Liferay data source. This is used in conjunction with session-factory.

The **session-factory** value specifies the session factory that is set to the persistence class. The default value is the Liferay session factory. This is used in conjunction with data-source.

The **tx-manager** value specifies the transaction manager that Spring uses. The default value is the Spring Hibernate transaction manager that wraps the Liferay data source and session factory.

If the **local-service** value is true, then the service will generate the local interfaces for the service. The default value is false.

If the **remote-service** value is true, then the service will generate remote interfaces for the service. The default value is true.

You can use the local-service and remote-service attribute according to your needs.

You are now connected with your external DB.

Put your Suggestions/Comments.

Thanks for Reading !!!

Posted by Amit Doshi at 1:16 PM

G+1 +6 Recommend this on Google

Labels: db + liferay, different database in liferay, external db in liferay

9 comments:

Ajay Saharan July 24, 2013 at 7:02 AM

We are using different database connection. We are able to connect with this db successfully. But when we try to fetch with primary key fields like FooLocalServiceUtil.getFoo(1)); we are able to get results.
But when we try to fetch by any other finder add by us in entity it's not giving results.

Reply

Amit Doshi   July 25, 2013 at 6:40 AM

It should be, we are implemented the above code in my project and it is live now.

Please check once again from your end each and every entry of ext-spring.xml file and let me know if you still find any issue.

Reply

StationI Liferay July 29, 2013 at 5:56 AM

Does liferay create tables in the different database which we mapped through the data-source(here testE) or to the Liferay's default database(here liferay612)?

Reply

Replies

Amit Doshi   August 2, 2013 at 12:37 AM

No. It will not create table in external schema. You need to manually create it with the sql file that was generated with the help of Service Builder

Reply

Aneal November 14, 2013 at 1:21 AM

I have implemented multitenant with shards. Everything is fine exception transaction management. The transaction is not rolled back in multitenant shards. However, it is rolled back in case of without murltitenat shards. Does using multitenant shards changes properties of Transaction manager? How can this be solved?

```
#Spring configuration files to be loadded. By adding shard-data-source-spring.xml in the list
database sharding feature
#can be enabled
spring.configs=\
META-INF/base-spring.xml,\
\
META-INF/hibernate-spring.xml,\
META-INF/infrastructure-spring.xml,\
META-INF/management-spring.xml,\
\
META-INF/util-spring.xml,\
\
META-INF/jpa-spring.xml,\
\
META-INF/executor-spring.xml,\
\
META-INF/audit-spring.xml,\
META-INF/cluster-spring.xml,\
META-INF/editor-spring.xml,\
META-INF/jcr-spring.xml,\
META-INF/ldap-spring.xml,\
META-INF/messaging-core-spring.xml,\
META-INF/messaging-misc-spring.xml,\
META-INF/mobile-device-spring.xml,\
META-INF/notifications-spring.xml,\
META-INF/poller-spring.xml,\
META-INF/rules-spring.xml,\
META-INF/scheduler-spring.xml,\
META-INF/scripting-spring.xml,\
META-INF/search-spring.xml,\
META-INF/workflow-spring.xml,\
\
META-INF/counter-spring.xml,\
META-INF/mail-spring.xml,\
META-INF/portal-spring.xml,\
```

```
META-INF/portlet-container-spring.xml,\
META-INF/staging-spring.xml,\
META-INF/virtual-layouts-spring.xml,\
META-INF/shard-data-source-spring.xml

shard.selector=com.liferay.portal.dao.shard.ManualShardSelector
shard.available.names=default,one,two
```

Reply

Tiago Augusto Coelho March 20, 2014 at 10:44 AM

Amit, I followed the post but when I perform the deploy I get the error ClassNotFoundException for hibernate classes. So I tried putting the jar but I get the error of ClassCastException. Just using eclipse wtp and I could test the portlet.

Reply

Mai Văn Quyết September 8, 2015 at 7:14 PM

So, i want to connect to two database, it's useful for me, please explain to me

Reply

rahuljeri January 28, 2016 at 10:29 PM

Nice blog. Visit our newly designed property portal website in india

Reply

Anna January 29, 2016 at 2:07 AM

Great and Useful Article.

Online Java Training | Online Java Training

Reply

Enter your comment...

Comment as: Reeshu Patel (Go   Sign out

Publish   Preview   ☐ Notify me

Newer Post   Home   Older Post

Subscribe to: Post Comments (Atom)

nRelate Posts Only

Blog Archive

► 2013 (4)

▼ 2012 (8)

  ► December (1)

  ▼ May (7)

    Dynamic Query in Liferay

    Deploying Ext in Liferay 6

    Connecting to different database using Liferay Ser...

    Redirect to View or Render Method after Submitting...

    LDAP Configuration with Liferay

    Liferay Eclipse Setup

    Access Liferay Service & Util and Custom Portlet ...

There was an error in this gadget

About Me

Amit Doshi

G+ Follow   42

By profession, I am working as a Senior CMS Consultant with CIGNEX DATAMATICS, Ahmedabad. I am Having around 4+ years of rich experience working and providing solutions for Java & Liferay . I've got recognition by Liferay Organization with Liferay Contributor 2012 Award. Check out Liferay Website: http://www.liferay.com/ community/special-projects/contributor-awards-2012. Amit Doshi has been involved with the Liferay Community since 2010. He initially started with version 5.2 and is now using Liferay 6.1. He is a member of Liferay India User Group(LIUG). He loves the open source concept and always admires it. He is always ready to help & contribute to the Liferay

community via forums or directly with individual whenever needed. He has also started a Liferay blog, to which he is constantly adding useful stuff regarding Liferay. He always trys to learn new things out of Liferay. He is leading Liferay projects as a Team Lead. He has completed his Bachelors in Computer Science from Saurashtra University in 2008. Amit Doshi has worked in many different kinds of Liferay projects in various industries such as Healthcare, Intranet Portal, Educational Portal, Media, etc.
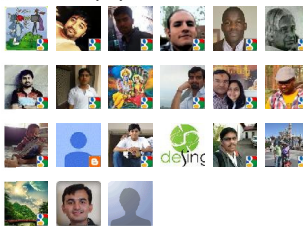
View my complete profile

About Me

Total Pageviews

128,382

Followers

Followers (35) Next

Follow

Simple template. Powered by Blogger.